Zakład Produkcji Oprogramowania VOTRON

ul.Klaudyny 16/136, 01-684 Warszawa

A. Ziółkowski

GENERATOR RAPORTOW BAZY DANYCH VOTRON

Warszawa 1987

1. Wprowadzenie.

Generator raportów RAP jest interpreterem specjalizowanego języka przeznaczonego do tworzenia wydruków o dowolnej postaci na podstawie danych przechowywanych w zbiorach bazy danych VITRIN. Dostępny w bazie danych VITRIN , program DRUK umożliwia sporządzanie raportów na podstawie danych zawartych w jednym zbiorze (ewentualnie z wykorzystaniem zbioru pomocniczego do wyszukiwania rekordów) przy czym raporty te mają ściśle określoną postać. Pisanie programów drukowania raportów w FORTRANIE jest bardzo uciążliwe. Program RAP umożliwia sporządzanie raportów na podstawie informacji zawartej w wielu zbiorach nie stwarzając ograniczeń na postać wydruków. Pisanie i późniejsze modyfikowanie programów drukujących raporty jest bardzo proste. Język umożliwia także realizację złożonego przetwarzania danych i dlatego może również zastąpić program LACZ używany w bazie danych VITRIN.

Program w języku interpretowanym przez RAP składa się z prostych instrukcji wykonywanych w naturalnej kolejności. Instrukcja **gdy** umożliwia warunkowe wykonywanie pewnych części programu a instrukcja **dopoki** wielokrotne powtarzanie.

Omawiając budowę instrukcji będziemy korzystali z następujących oznaczeń:

[ ] - nawiasy kwadratowe służą do oznaczenia części opcjonalnej instrukcji,

[¦] - kreski pionowe służą do oznaczenia alternatyw (jedna z części musi wystąpić w instrukcji),

< > - nawiasy ostre służą do oznaczenia parametru formalnego,

[ ]... – nawiasy kwadratowe z kropkami służą do oznaczenia części instrukcji, która może wystąpić wielokrotnie.

2. Instrukcje dotyczące zbiorów.

otworz <zb>,<sek> [:<nzb>]

Instukcja **otworz** powoduje odszukanie i otwarcie zbioru o podanej nazwie <zb> na sekcji <sek> . Jednocześnie można otworzyć wiele zbiorów. Ograniczenie ilości jednocześnie otwartych zbiorów wynika jedynie z wielkości dostępnej pamięci. Opcjonalny parametr <nzb> – nowa nazwa zbioru, służy do zdefiniowania nazwy zbioru, która będzie używana w odwołaniach do zbioru w dalszej części programu. Zbiory otwarte jednocześnie muszą mieć różne <nzb>. Pominięcie tego parametru oznacza, że przyjęto <nzb> równe <zb>. W przypadku, gdy zbiory otwierane na różnych sekcjach mają te same nazwy, <nzb> przynajmniej dla jednego zbioru musi być podane. Po otwarciu zbioru dostępne są wszystkie dane występujące w dokumencie (rekordzie).

czytaj <nzb>

Instrukcja wczytania następnego dokumentu.

czytaj <nzb> : <dana> [,<dana>]...

Instrukcja wczytania dokumentu o podanym identyfikatorze. Dane podane po : są kolejnymi danymi wchodzącymi w skład identyfikatora.

czytaj <nzb> = [<liczba>|<zmienna>]

Instrukcja wczytania dokumentu o podanym numerze fizycznym. Znaczenie parametru <zmienna> omówiono w dalszej części

opracowania.

pisz <nzb>

Instrukcja pisania do kolejnego dokumentu. Instrukcja nie zmienia identyfikatora. Osiągnięcie końca zbioru sygnalizowane jest jako błąd (zmienna $bl różna od zera).

pisz <nzb> : <dana> [,<dana>]...

Instrukcja pisania dodokumentu o podanym identyfikatorze. Jeżeli dokument jeszcze nie istnieje to zostanie utworzony.

pisz <nzb> = [<liczba>¦<zmienna>]

Instrukcja pisania do dokumentu o podanym numerze fizycznym. Jeżeli dokument o tym numerze jeszcze nie istnieje, sygnalizowany jest błąd (zmienna $bl różna od zera).

usun <nzb>

Instrukcja oznaczania kolejnego dokumentu jako usuniętego.

usun <nzb> : <dana> [,<dana>]...

Instrukcja oznaczania dokumentu o podanym identyfikatorze jako usuniętego.

usun <nzb> = [<liczba>¦<zmienna>]

Instrukcja oznaczania dokumentu o podanym numerze fizycznym jako usuniętego. Podczas usuwania sygnalizowany jest błąd gdy usuwany dokument nie istnieje.

3. Dane, nazwy danych, parametry i zmienne.

Przy definiowaniu raportu lub algorytmu przetwarzania można odwoływać się do następujących wielkości:

- danych z dokumentów,

- parametrów ogólnych,
- parametrów zbioru,
- zmiennych pomocniczych.

Odwołnie do danej z dokumentu ma postać:

<dana> [.<nzb>]

gdzie <dana> jest kodem danej a <nzb> nazwą zbioru zdefiniowaną instrukcją otworz. Nazwa zbioru wraz z poprzedzającą kropką jest częścią opcjonalną. Gdy zostanie pominięta przyjmowana jest domyślnie nazwa zbioru z ostatniego odwołnia. Ponieważ można otworzyć wiele zbiorów, w których mogą występować dane o tych samych nazwach, podawanie nazwy zbioru w odwołaniu jest w ogólnym przypadku konieczne.

Odwołanie do pełnej 28-mio znakowej nazwy danej ma postać:

@<dana> [.<nzb>]

Wszystkie odwołania do parametrów oraz zmiennych rozpoczynają się znakiem $, po którym podawana jest nazwa parametru. Wyróżniono następujące parametry ogólne:

$bl - numer błędu. Zero oznacza poprawne wykonanie ostaniej instrukcji.

$ad - aktualny dzień (1..31)

$am - aktualny miesiąc (1..12)

$ar - aktualny rok (1987..)

$ns - numer strony

$nw - numer wiersza

Odwołania do parametrów zbioru mają postać:

$<par> [.<nzb>]

gdzie <par> jest nazwą parametru a <nzb> podawaną opcjonalnie nazwą zbioru. W przypadku W przypadku pominięcia nazwy zbioru przyjmuje się domyślnie, że odwołanie dotyczy zbioru do którego się ostatnio odwoływano. Można odwoływać się do wszystkich parametrów zbioru, używając nazw podanych w dokumentacji systemu VITRON np. $ald, $dklu, $ngen. Ponadto dla każdego zbioru wprowadzono dodatkowo parametr:

$nr - aktualny numer dokumentu. Parametr ten przyjmuje wartość zero gdy w ostatnio wykonywanej instrukcji czytania, pisania lub usuwania dokumentu wystąpił błąd.

Do wykonywania obliczeń i przechowywania informacji można używać zmiennych o następujących nazwach:

Zmienne całkowite:

$a

$b

...

$z

Zmienne rzeczywiste:

$aa

$bb

...

$zz

Zmienne tekstowe:

$aaa

$bbb

...

$zzz

4. Instrukcje podstawienia i kopiowania.

Instrukcja podstawienia ma identyczną postać jak w języku FORTRAN. Po lewej stronie znaku podstawienia = może wystąpić dana lub zmienna, po prawej wyrażenie arytmetyczne, w którym oprócz danych i zmiennych mogą występować parametry oraz stałe liczbowe. Do zpisywania wyrażeń arytmetycznych używa się operatorów działań:

+ - dodawanie,

- - odejmowanie,

* - mnożenie,

/ - dzielenie,

oraz nawiasów okrągłych (i).

Oprócz instrukcji podstawienia wprowadzono instrukcję kopiowania wartości tekstowych, którą zapisuje się podając daną lub zmienną, na którą tekst będzie kopiowany, a po znaku : określa się tekst, który ma być skopiowany. Może to być:

@<dana> [.<nzb>] - 28-mio znakowa nazwa danej,

$<par> [.<nzb>] - parametr znakowy np. nazwa zbioru,

"<tekst>" - stała tekstowa,

$<zmienna> - zmienna tekstowa, np. $aaa, $ttt

<dana> [.<nzb>] - dana tekstowa (o formacie A)

Operator konkatenacji + umożliwia łączenie tekstów pochadzących z różnych źródeł w jedną całość. Ponadto można określić numer znaku, od którego ma nastąpić kopiowanie, podając

po znaku ' odpowiednią warość.

Przykłady:

$a = $a + 1

Zwiększenie wartości zmiennej całkowitej $a o 1.

wyn.wy=(war.zb-$aa)/(war+$aa)

Obliczenie wartości wyn w zbiorze wy określonej za pomocą wyrażenia arytmetycznego na podstawie wartości danej war ze zbioru zb oraz zmiennej rzeczywistej $aa.

$aaa : "Odbiorca:" + nodb.zb'6

Kopiowanie do zmiennej tekstowej $aaa tekstu "Odbiorca:" oraz bezpośrednio po nim tekstu z danej nodb ze zbioru zb od 6-tego znaku. (Uwaga - długość tekstu zmiennej nie może przekraczać 100 znaków).

5. Gdy, dopóki.

Instrukcja **gdy** umożliwia warunkowe wykonywanie innych instrukcji. Ogólna postać tej instrukcji jest następująca:

gdy <wyrażenie logiczne> : <instrukcja>

W wyrażeniu logicznym mogą wysrępować dane, parametry, zmienne i stałe. Do zapisywania wyrażeń arytmetycznych używa się operatorów:

& - iloczyn logiczny,

! - suma logiczna,

> - większy,

= - równy,

- różny,

< - mniejszy,

oraz nawiasów okrągłych (i).

Relacje większy, równy, różny, i mniejszy odnoszą się do wielkości całkowitych, rzeczywistych i tekstowych.

Występująca po znaku : instrukcja może być zastąpiona blokiem (sekwencją) instrukcji ograniczonym nawiasami { i } (lub << i >>).

Przykłady:

```
gdy wyn.zb < 0 : wyn.zb = 0
gdy wyn.zb = 0 & id # "###" :
 {
 czytaj  we : id
 wyn.zb = wyn.we
 pisz  zb : id
 }
```

Instrukcja **dopoki** umożliwia wielokrotne wykonywanie innych instrukcji. Ogólna postać tej instrukcji jest następująca:

dopoki <wyrazenie logiczne> : <instrukcja>

Podobnie jak w instrukcji **gdy** występująca po znaku : instrukcja może być zastąpiona blokem instrukcji w nawiasach { i }

Przykład:

```
$ss = 0
otworz  zb.asa = we
dopoki  $nr.we # 0 :
 {
 czytaj  we
 $ss = $ss + war.we
 }
```

6. Wiersz, kolumna.

Przyjęto następującą zasadę sporządzania raportów. Wszystkie wydruki są wstępnie przygotowywane w buforach o pojemności 1 wiersza drukarki (160 znaków). Bufory te nazywane w skrócie wierszami są ponumerowane 1,2,3... itd. Bufory mogą być podzielone na kolumny, których szerokość jest definiowana w programie. Kolumny są również ponumerowane 1,2,3...itd. Aby wprowadzić informację do bufora określa się zawsze numer wiersza oraz numer kolumny. Po przygotowaniu informacji do wydruku w buforach, zawartość buforów o podanych numerach jest wyprowadzana na drukarkę lub ekran monitora.

Wiersze oraz kolumny są definiowane instrukcją wzor Instrukcja ta służy również do wstępnego wypełniania bufora potrzebnymi znakami. Ogólna postać instrukcji wzor jest następująca:

wzor <wiersz> [,<kolumna> ("<znaki pocz.>",<szer.kolumny>,
"<znaki kon.>")]...

W wyniku działania instrukcji zostanie utworzony bufor wiersza o numerze **<wiersz>** podzielony na kolumny o podanych numerach **<kolumna>** , które zostaną wstępnie wypełnione znakami początkowymi oraz znakami końcowymi. Do dalszego wypełniania kolumny zostanie zarezerwowana ilość znaków określona parametrem <szer.kolumny> .

Jedną instrukcją wzór można definiować kilka buforów o kolejnych numerach wiersza. W tym przypadku zamiast jednego parametru **<wiersz>** podaje się dwa parametry oznaczające numer pierwszego i ostatniego wiersza oddzielone kropkami. W podobny

sposób można określać kolumny o identycznym wzorcu.

Przykład:

wzor 1..4,1("I ",20," "),2..4("I ",10," ")

Parametr <szer.kolumny> ma następującą postać:

[,<liczba>]...

[,<dana> [.<nzb>]]...

[,<wiersze> : @<dana> [.<nzb>]]...

[,<wiersze> : "<tekst>"]

Każda z 4 opcji definiuje w inny sposób szerokość kolumny. W jednej definicji można użyć więcej niż jedną opcję. Również opcje tego samego typu mogą występować wielokrotnie. Przy takiej definicji zostanie przyjęta największa szerokość kolumny.

Pierwsza opcja definiuje bezpośrednio szerokość kolumny w znakach. Druga opcja służy do zarezerwowania miejsca do wydruku danej ze zbioru. Szerokość kolumny przyjmowana jest na podstawie parametrów zbioru określających format tej danej. Dwie ostatnie opcje służą do wypełniania nagłówka odpowiednim tekstem (28-mio znakową nazwą danej lub podanym bezpośrednio tekstem). Powodują one wpisanie tekstu do buforów wierszy o numerach (początkowym i końcowym), przy czym tekst zostanie tak rozmieszczony w poszczególnych wierszach (bez dzielenia słów) aby kolumna była możliwie wąska. W podobny sposób jest rozmieszczany tekst nazw danych w nagłówkach programu DRUK.

Szerokość kolumny może być zdefiniowana w programie tylko raz. W następnych instrukcjach wzor można podawać dla danej kolumny tylko znaki początkowe i końcowe.

7. Wypełnianie buforów wiersza.

Do wypełniania buforów wiersza służy instrukcja:

wiersz <wiersz> [[,<kolumna> ['<format>] = <wyr.arytm.> |

,<kolumna> ['<nr znaku>] : <wyr.tekstowe>]]...

Instrukcja umożliwia wpisanie wartości wyrażenia arytmetycznego, zgodnie z podanym formatem, do bufora wiersza o numerze **<wiersz>** i kolumny o numerze <kolumna> Format określa się podając liczbę znaków przed kropką dziesiętną oraz po znaku . liczbę znaków po kropce. Dla zmiennych całkowitych nie podaje się ostatniego parametru i poprzedzającego go znaku.

Instrukcja **wiersz** umożliwia ponadto kopiowanie tekstu od znaku kolumny równego **<nr znaku>** lub od pierwszego znaku gdy nie podano tego parametru. Zasady pisania wyrażeń arytmetycznych są identyczne jak w omawianych wcześniej instrukcjach podstawienia.

Bezpośrednio po znaku ' można podać jeden ze znaków określających opcje:

> - przesunięcie wpisywanego tekstu maksymalnie w prawo,

< - przesunięcie wpisywanego tekstu maksymalnie w lewo,

* - umieszczenie tekstu na środku,

$ - drukowanie wartości ze spacją co 3 znaki licząc od kropki (np. '$5.3 daje wydruk w postaci 99 999.999).

Przykład:

wiersz 1, 1 '* : "Wynagrodzenia pracownikow"

wiersz 5, 2 : nazw, 3 = wynagr, 4 '5.3 = wynagr + prem

8. Drukowanie raportów.

Drukowanie raportu polega na wyprowadzaniu zawartości buforów wierszy o podanych numerach. Służy do tego instrukcja drukuj :

drukuj <wiersz> [,<wiersz>]...

Wszystkie wiersze podane w jednej instrukcji są drukowane na tej samej stronie.

Instrukcja **naglowek** w postaci:

naglowek <liczba wierszy na stronie> : <wiersz> [,<wiersz>]...

[/ <wiersz> [,<wiersz>]...]

[: <instrukcja>]...

określa numery wierszy, które należy wyprowadzić na początku następnej strony w momencie gdy wartość parametru $nw - numer wiersza przekroczy podaną wartość <liczba wierszy na stronie> . Opcjonalnie można podać (po znaku /) numery wierszy, które należy wydrukować na końcu każdej strony (tzw. stopka).

Parametr **$nw** jest automatycznie powiększany przez instrukcję **drukuj** . Wartość parametru **$nw** może być zmieniana w programie gdy zachodzi taka potrzeba. Wstępnie ustawiana duża wartość parametru **$nw** powoduje wydrukowanie nagłówka na początku pierwszej strony wydruku po wywołniu pierwszej instrukcji **drukuj** . Dlatego jeżeli trzeba umieścić jakiś tekst przed nagłówkiem należy zmienić wartość tego parametru.

Przy określaniu nagłówka można używać wiersza o numerze 0 wypełnionego z definicji znakami minus.

W instrukcji **naglowek** można również określić po znaku : instrukcję lub blok instrukcji, które należy wykonać przed wyprowadzeniem kolejnego nagłówka na nowej stronie. W większości przypadków będzie to instrukcja wpisania do bufora jednego z wierszy nagłówka warości parametru $ns - numer strony.

9. Parametryzacja wydruków.

Programy pisane w języku generatora raportów RAP mogą być parametryzowane. Instrukcja **parametr** w postaci:

parametr %<numer param.> : <nazwa param.> : <wartosc param.>

powoduje wyświetlenie na ekranie nazwy i po znaku : wartości parametru. Użytkownik może zachować wyświetloną wartość parametru wprowadzając pusty rekord lub wprowadzić nową wartość parametru. We wszystkich miejscach programu, w których wystąpi tekst:

%<nr parametru>

zostanie on zastąpiony tekstem **<wartosc parametru>**.

Przykład:

```
parametr %1 :numer kwartalu : 1
...
czytaj  zb
gdy  nrkw = %1
 {
 ...
 }
```